REMARKS/ARGUMENTS

The Examiner is thanked for the performance of a thorough search. A substitute Specification has been submitted to denote JAVA as a registered trademark. By this amendment, Claim 1-9, 12-27, 31-36, 38-45, 47-51 and 54-59 has been amended. No claims have been added, or cancelled. Hence, Claims 1-63 are pending in the application. Each issue raised in the Office Action mailed January 27, 2005 is addressed hereinafter.

THE REJECTIONS NOT BASED ON PRIOR ART

The Specification and drawings were objected due to the use of the trademark JAVA. The Office Action states that the use of the trademark JAVA should be capitalized wherever it appears and be accompanied by the generic terminology. The Applicant respectfully requests approval of the substitute Specification as depicted in the Appendix accompanying this paper. The Appendix includes both a red line version of the specification, each page of which is labeled "Annotated Sheet Showing Changes," as well as new sheets that are labeled "Replacement Sheet" and which include a clean copy of the amended Specification. The proposed changes have been submitted to include proper capitalization of the trademark JAVA. No new matter is introduced by the substitute specification. Consideration is respectfully requested.

THE REJECTIONS BASED ON THE PRIOR ART

Claims 1 – 63 were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,535,894 B1 issued to Schmidt et al. (herein referred to as "Schmidt"). Specifically, The Office Action states that the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been

obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains.

The Office Action does not identify **any support** in **the reference** for the assertion that the claimed invention would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. As shall be explained in detail hereafter, Schmidt is fundamentally different than Applicant's claimed invention. Due to the fundamental differences it would not be obvious to one skilled in the art that Applicant's claimed invention was an obvious variation of Schmidt.

Claims 1 as amended recites:

> A system for providing a JAVA code release infrastructure with granular code patching, comprising:
>
> one or more JAVA code patches, each comprising at least one resource unit, each resource unit comprising metadata and file components;
>
> one or more JAVA code libraries, each comprising at least one such resource unit;
>
> a patch tool, comprising:
>
> a compare module that determines which units in the JAVA Code libraries are out of date by comparing the metadata for each such resource unit in the JAVA code patches to the metadata for each such corresponding resource unit in the JAVA code libraries; and
>
> a merge **module merging each such resource unit in the JAVA code patches into the JAVA code libraries** for each such corresponding resource unit that is out-of-date.

The Office Action states that the "one or more JAVA code patches, each comprising at least one resource unit" as recited in Claim 1 correlates to the target archive file and difference archive file described by Schmidt. (col. 5, lines 44-48) This is incorrect. The target archive file and difference archive files are fundamentally different from the JAVA code patches and cannot possibly correspond to the JAVA code patches recited in Claim 1.

THE TARGET ARCHIVE FILE IS NOT THE CLAIMED CODE PATCH

Specifically, the target archive file described in Schmidt cannot correspond to the JAVA Code Patch recited in Claim 1 because in Schmidt no resource units of the **target archive file** are ever merged into the JAVA Code Libraries that are updated. The target archive file remains at all times on the Remote server and does not directly interact with the to-be-updated archive. (See Schmidt FIG. 4 and FIG. 7) This is fundamentally different from the JAVA code patch, which is the actual component merged with the JAVA Code libraries. (See Application FIG. 1)

In contrast to Schmidt's target archive file, Claim 1 recites that "a merge module merging each such resource unit in the **JAVA code patches into the JAVA Code Libraries** for each such corresponding resource unit that is out-of-date." The target archive file described in Schmidt is not the component applied to the to-be-updated archive files. (See Schmidt FIG. 7 and associated text) The fact that the target archive file is not directly applied to the to-be-updated archive files means that the target archive file cannot possible correlate to the JAVA Code patches recited in Claim 1, which is expressly merged into the JAVA Code Libraries during a patch operation.

THE DIFFERENCE ARCHIVE FILE IS NOT THE CLAIMED CODE PATCH

The difference archive file described in Schmidt also does not correlate to the JAVA Code patches described in Claim 1 because which resource units are out of date is not determined by comparing metadata in the difference archive with metadata in the "original archive". In fact, the difference archive file already reflects which resource units are out of date as a result of comparisons between the target file and original file. (See FIG. 11 and associated text) This is fundamentally different from the JAVA code patch recited in Claim 1.

Specifically, Claim 1 states a "compare module that determines which resource units in the JAVA code libraries are out of date by **comparing the metadata for each such resource unit in the JAVA Code patches** to the metadata for each such corresponding resource unit in the JAVA code libraries." Therefore, the difference archive file, which already reflects the out of date portions of an archive, cannot correlate to the JAVA Patch code which is used to determine the out of date resource units in the JAVA Code Libraries.

THE COMPARE MODULE

An essential difference from Schmidt is the compare module recited in Claim 1, which is use to determine which units in the to-be-patched JAVA Code Libraries are out of date. The comparison is done by comparing metadata in the to-be-patched JAVA Code Libraries with the metadata in the JAVA Code Patch. The comparison is done by looking at the specific Java code libraries that are to be patched. The comparison performed by the claimed invention significantly teaches away from Schmidt, which does not involve the to-be-patched archives when determining what content within the archives is out of date.

DETERMINING WHAT IS OUT-OF-DATE

To determine what is out-of-date, Schmidt requires that the server maintain an exact copy of what the clients' current archive file looks like in order to generate the difference archive. If one does not know what the clients' current archive looks like, then one cannot properly update the clients' files using Schmidt's technique. Additionally, only client archive files that are identical to Schmidt's "original archive" will be updated properly by the difference archive file. Clients' archive files which have different out of date files cannot be updated by the same difference archive file. Because Schmidt performs, on the **server side,** the

comparison (between the original archive and the target archive) to determine which units are out of date, without looking to see what each client's files actually look like, the Schmidt technique differs fundamentally from the invention recited in Claim 1.

The invention recited in Claim 1 on the other hand, performs **the comparison to determine the out of date files** against the actual code libraries that are being patched. Thus, there is no need to know, prior to generating the path, the exact state of each to-be-patched library. Additionally the same patch will work for multiple clients, all of which have different out of date code libraries.

OTHER CLAIMS

Claims 19 and 37 are identical to independent Claim 1 except that Claim 19 is a method claim and Claim 37 is a computer readable medium claim. Each of Claims 19 and 37 is therefore allowable for the reasons give above with respect to Claim 1.

The remaining dependent claims depend upon one of the independent claims discussed above, and thus include each and every feature of the corresponding independent claims. Each of the remaining dependent claims is therefore allowable for the reasons given above for Claims 1, 19 and 37. In addition, each of remaining dependent claims introduces one or more additional limitations that independently render it patentable. However, due to the fundamental differences already identified, to expedite the positive resolution of this case a separate discussion of those limitations is not included at this time.

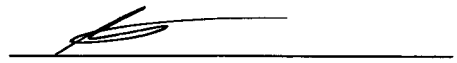
For the reasons set forth above, it is respectfully submitted that all of the pending claims are now in condition for allowance. Therefore, the issuance of a formal Notice of Allowance is believed next in order, and that action is most earnestly solicited.

The Examiner is respectfully requested to contact the undersigned by telephone if it is believed that such contact would further the examination of the present application.

Please charge any shortages or credit any overages to Deposit Account No. 50-1302.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP



Brian D. Hickman
Reg. No. 35,894

2055 Gateway Place, Suite 550
San Jose, CA 95110-1089
(408) 414-1080
Date: April 6, 2005
Facsimile: (408) 414-1076

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Mail Stop Amendment, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450

on 4/6/05 by Darci Sakamoto